Hierarchical Text Categorization as a Tool of Associating Genes with Gene Ontology Codes

Svetlana Kiritchenko
University of Ottawa
P.O. Box 450 Stn A,
Ottawa, Ontario
K1N 6N5 Canada
svkir@site.uottawa.ca

Stan Matwin
University of Ottawa
P.O. Box 450 Stn A,
Ottawa, Ontario
K1N 6N5 Canada
stan@site.uottawa.ca

A. Fazel Famili
IIT NRC
1200 Montreal Rd.,
Ottawa, Ontario
K1A 0R6 Canada
Fazel.Famili@nrc-cnrc.gc.ca

ABSTRACT

A great deal of genomics information accumulated through years is available nowadays in on-line text repositories such as Medline. These resources are essential for biomedical researchers in their everyday activities on planning and performing experiments and verifying the results. However, these resources do not still provide adequate mechanisms for retrieving the requisite information. We propose a new tool for assisting biologists with literature search for the task of associating genes with Gene Ontology codes. Unlike previous research, we design the hierarchical text categorization framework to address this problem. The hierarchical approach helps visualize the results, address the scalability issue, improve classification accuracy and trade off between precision and recall.

1. INTRODUCTION

In many genomics studies one of the major steps is the gene expression analysis using high-throughput DNA microarrays. Measuring the expression profiles of genes from normal and disease tissues or from the same tissue exposed to different conditions can help discover genes responsible for the disease. It can also shed light on the functionality of previously unknown genes. Traditionally, most computational research on analyzing gene expression data has focused on working with microarray data alone, using statistical or data mining tools. However, raw gene expression data are very hard to analyze even for an experienced scientist. On the other hand, there exists a wealth of information pertaining to the function and behavior of genes, described in papers and reports. Most of these are available on-line and could potentially be useful in the analysis of gene expression, if we had a way of harvesting this information and combining it synergetically with the knowledge acquired from the microrray data experiments. Specifically,

our research is aimed at providing molecular biologists with known functional information on genes used in the experiments in order to make microarray results and their analysis more biologically meaningful.

Another important aspect of any genomics study is the validation step. To become widely accepted, new discoveries have to be validated by other experiments or confirmed by related research. A common practice for validation is to check scientific literature for similar results. For example, suppose that in an Alzheimer's study several genes were identified as highly related to the disease. Then, the literature search of related research showed that some of these genes have already been known as associated with other neurological disorders. This fact would be a supporting evidence for the results of the Alzheimer's study. However, such validation requires extensive literature search, which is most often done manually. Automatic text analysis techniques can effectively replace manual effort in this area.

Even though many genes for well-studied organisms, such as Escherichia coli or Saccharomyces cerevisiae, have been already annotated in specialized databases (EcoCyc, SGD), information on many other genes currently can be found only in scientific publications. Public databases are created and curated manually; thus, they cannot keep up with an overwhelming number of new discoveries published on a daily basis. Furthermore, these databases often use different vocabularies to describe gene functionality, which raises an additional challenge for integrating the results. Consequently, genomics databases are not always adequate to find the requisite information. Therefore, we need to apply text mining and categorization techniques to retrieve up-to-date information from biomedical literature and translate it into a standardized vocabulary to help life scientists in their everyday activities. At the same time, the same process can be used as a tool to assist in updating and curating databases.

In machine learning literature, the standard text classification framework is described as follows: given a domain of documents D and a set of predefined categories $C = \{c_1, \ldots, c_{|C|}\}$, the task of text categorization is to assign a Boolean value to each pair $\langle d_j, c_i \rangle \in D \times C$ [6]. Let us observe, however, that this framework does not match the task of classifying genomics data, as the ontologies into which we want to classify are not just flat sets of categories, but hierarchies by their nature. Therefore, we must turn to the hierarchical text categorization framework to realize

the goal of this research. An immediate observation is that unlike the most used, flat text classification framework, the area of hierarchical classification has received little attention. We would like to fill in this gap and bring the benefits of hierarchical text categorization to genomics in general and gene function identification in particular.

2. RELATED WORK

The text-related approach to functional annotation of genes in general corresponds to classifying articles describing a particular gene into one or several functional categories. Then, the discovered categories are assigned to the gene. The straightforward technique of classifying Medline abstracts into Gene Ontology (GO) terms using standard machine learning algorithms (maximum entropy, Naive Bayes, and nearest neighbour) was proposed by Raychaudhuri et al. [5] and showed promising results. After each article associated with a gene has been classified into GO terms, the gene function was chosen based on a weighted voting scheme. Similar approach was used in Euclid system [9]. They used a simple keyword-based classifier to assign functions to sequences from Swiss-Prot database based on the detailed free-text functional annotations provided by human experts.

Several researchers addressed the problem of verifying if a given set of genes shares a function [7, 4]. This task is especially important in microarray analysis. The usual practice in microarray analysis is to cluster genes by their expression profiles. If genes in a cluster share functionality, the cluster is considered interesting and would be a good candidate for the follow-up studies. Shared functionality verification is based on the assumption that genes share functionality if the articles describing these genes share the content. So, researchers looked at the sets of articles similar in content to the ones that describe a particular gene and see how many of them are common for genes in a set. Masys et al. [3] applied a similar idea to keywords assigned to documents describing a given set of genes. In the Medline database articles are indexed with terms from biological ontologies such as Medical Subject Headings (MeSH) or Enzyme Commission (EC) codes. Masys et al. looked for the MeSH and EC terms that appear frequently for a set of genes and visualized a term hierarchy with genes whose documents were assigned the corresponding terms.

Our application closely follows the ideas in [5] with the exception of hierarchical classification. The goal of their work was similar to ours: to assign biological functions to genes by classifying the Medline articles associated with the genes. They applied conventional machine learning techniques to the biological texts classifying them into a flat set of categories. For their experiments they chose only 21 categories and used no hierarchical information. We are going to extend their approach and employ the whole GO hierarchy applying hierarchical categorization methods to the problem. The hierarchical techniques allow us to address the scalability issue of the application.

3. HIERARCHICAL FUNCTIONAL ANNOTATION OF GENES

3.1 Associating genes with GO codes using text categorization
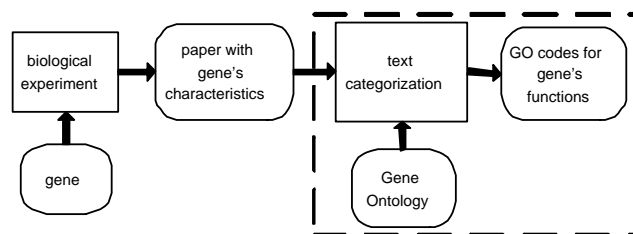
Figure 1: Functional annotation process. Genes' functions are determined in biological experiments and stated in scientific publications. The goal of our task (in the dashed box) is to retrieve these functions and translate them into corresponding GO codes.

In this work, we propose a system to classify genes/gene products into Gene Ontology codes based on the classification of documents from the Medline library that describe the genes. The purpose of this task is to retrieve the known functionality of a group of genes from the literature and translate it into a controlled vocabulary (see Figure 1). Our categories come from Gene Ontology [1]. In biology controlled vocabularies for different subdomains are traditionally designed in the form of ontologies [8]. Gene Ontology is quickly becoming a standard for gene/protein function annotation, and so, it is our choice for the hierarchy of categories.

Several databases, such as SGD, MGD, etc., have information on genes of a particular species. For example, the Saccharomyces Genome Database (SGD) contains the sequences of yeast genes and proteins; descriptions and classifications of their biological roles, molecular functions, and subcellular localizations (GO codes); and links to literature information. We can use this information on well-studied genes to train a classification system in order to automatically fill in the gaps in annotations for more recently studied genes. We do this as follows.

Learning. We learn a classification system from the information on fully annotated genes in the database in the following steps:

1. retrieve the GO codes and the IDs of Medline articles associated with the genes from the database;

2. retrieve the corresponding articles from the Medline library;

3. form a training set consisting of the words from the retrieved articles as features and the corresponding GO codes as categories;

4. learn a classifier using hierarchical text categorization techniques;

Classification. After the classification system has been learned, we can apply it to genes that do not have annotations in the database:

1. retrieve all Medline articles that mention the genes (possibly using all their aliases to improve recall);

2. classify the retrieved articles into GO codes with the classifier;

3. assign the most promising codes to the genes (the most promising codes can be identified as the majority codes

from article classification or by using more complex techniques involving background knowledge).

For example, for yeast genes, from the SGD database we collect the IDs of Medline articles associated with the genes and Gene Ontology codes manually assigned to these genes. We retrieve the corresponding articles from the Medline library and form a training set consisting of articles and GO labels. We train a classification system to be able to classify any biomedical text mentioning a yeast gene into one or several GO codes. Then, suppose we do not have an annotation for gene YDR341C, cytoplasmic arginyl-tRNA synthetase. We retrieve all the Medline articles mentioning YDR341C or any of its aliases and classify them into GO codes. Our classification system should produce GO code $GO : 0006412$, protein biosynthesis, or at least any of its ancestor codes.

To evaluate the performance of our system we can use the cross-validation techniques or split the data into training and test sets where the test set would consist of more recent articles than the training set to simulate the use of the system in the real-life settings.

The classification process can be used for automatic or semi-automatic database curation and maintenance. At the same time, it can be used as a stage in the gene expression analysis. After the microarray experiments have been performed and the gene expression data have been preprocessed and clustered, the information on gene functions can be added as background knowledge. For some genes used in experiments the functional information is readily available from databases, yet for others, less studied genes we can use our scheme to retrieve the functional annotation from biomedical literature. We just have to retrieve all Medline documents talking about the gene and classify them to get the gene's functions as described above.

3.2 Hierarchical categorization of genes into the GO hierarchy

The described technique has been already in use by other researchers [5]. However, they treated Gene Ontology as a flat set of categories and, as a result, applied conventional machine learning algorithms ignoring the hierarchical relations of the GO codes. We, on the other hand, would like to explore all potential of the Gene Ontology hierarchy by utilizing hierarchical text categorization techniques.

Since the mid-1990s machine learning researchers have realized that many large text collections, such as web directories, patent databases or biomedical data, are organized hierarchically and, therefore, require special techniques to deal with. As a result, several learning methods have been developed to incorporate hierarchical information resulting in more accurate and faster classification.

Hierarchical text categorization methods can be divided in two types [10]: global (or big-bang) and local (or top-down level-based). A learning approach is called global if it builds only one classifier to discriminate all categories in a hierarchy. A global approach differs from flat categorization in that it somehow takes into account the relationships between the categories in a hierarchy. A learning approach is called local if it builds separate classifiers for internal nodes of a hierarchy. A local approach usually proceeds in a top-down fashion, first picking the most relevant categories of the top level and then recursively making the choice among the low-level categories, children of the relevant top-level categories.

A local approach seems natural for hierarchical classification since it reflects the way that humans usually perform such tasks. Going level by level, each time discriminating just a few categories is much easier than discriminating hundreds of categories at once. The same is also true for automatic systems. In machine learning it is known both theoretically and experimentally that the more categories are present in the classification task, the more difficult the task is, and therefore, the lower classification accuracy can be reached. Also, the intuition tells us that classifying into high-level categories is easier than discriminating among all categories not only because the number of categories is smaller but also because they are more distinctive. Then, after correct categorization into one of the high level categories is done, the number of possible low-level categories becomes smaller if we consider only the children of the correct high-level categories.

Since Gene Ontology consists of hundreds of categories, a straight-forward application of conventional "flat" classification methods will not produce an accurate system. Therefore, we decided to apply local hierarchical techniques to address such a large number of categories. More specifically, we apply a local hierarchical technique called Pachinko Machine. In a Pachinko Machine the classification decisions are made in a top-down fashion iteratively and irreversibly. At each level the classifier selects one (or several, in case of multi-label categorization) of the most probable categories for a test example and then proceeds down the hierarchy inspecting only the children of the selected nodes.

We propose to improve this method by allowing stopping earlier in the hierarchical graph. If a classifier at some level is unsure about which path to choose on the next step, we stop the classification process and assign the internal node as a final decision. In this manner, the classifier would possibly avoid going on the wrong path, yet its decision is not fully defined since we do not know which of the categories below in the hierarchy best describe the gene function.

We have designed a simple algorithm based on a local hierarchical approach of Pachinko Machine. The algorithm first builds a classifier for the categories of the first level of the hierarchy. Any machine learning method that outputs the probability scores[1] can be used here to build a classifier. This classifier then generates a set of scores for each new instance and only those categories for which an instance has a score greater than a given threshold are assigned to the instance. If none of the categories has got a score greater then the threshold, the instance is assigned to the root category and the categorization process stops. Otherwise, on the next step the learning process is repeated for all categories assigned to an instance classifying it into categories at deeper levels of the hierarchy.

Let us notice, however, that this method would produce very poor results in terms of standard evaluation measures: accuracy/error and precision/recall. All category assignments differing from the correct category would be considered equally erroneous. The conventional measures cannot differentiate between different kinds of misclassification errors and, therefore, are not suitable for hierarchical categorization. Intuitively, misclassification to a sibling or a parent node of the correct category is much better than misclassifi-

---

[1]A probability score can represent an actual probability of an instance belonging to a category or it can just reflect the ranking order of relevancy of instances to a category.
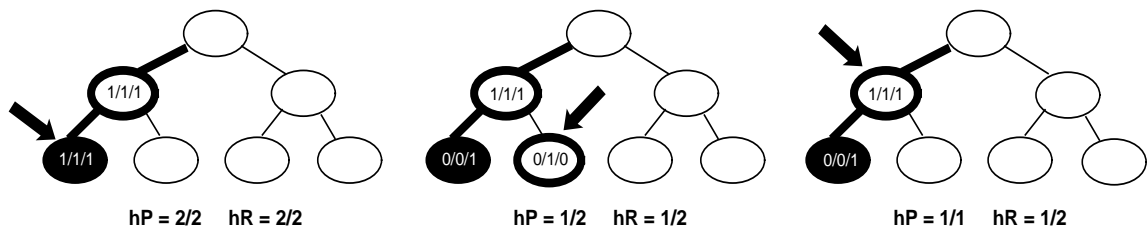
Figure 2: New hierarchical measure. The solid ellipse represents the real category of a test instance; the ellipse in bold with an arrow pointing to it represents the category assigned to the instance by a classifier; all nodes on the path from the root to the assigned category, ancestors of the category are shown in bold since they are also assigned to the instance by our measure. The path from the root to the real category, the correct path is shown in bold. Numbers in the nodes represent the number of instances correctly assigned to the node by the classifier (TP), total assigned by the classifier ($TP + FP$) and total assigned by the experts ($TP + FN$), respectively. Hierarchical precision hP and hierarchical recall hR shown for each case are microaveraged.

cation to a distant node. Therefore, we need a new measure that would be consistent with conventional non-hierarchical measures but more discriminating allowing us to give credit for less severe misclassification errors.

We propose a new hierarchical evaluation measure based on two principles:

1. to give higher (better) evaluation for correctly classifying one level down comparing to staying at the parent node;

2. to give lower (worse) evaluation for incorrectly classifying one level down comparing to staying at the parent node.

To satisfy these two principles, we have chosen our measure to be the pair precision and recall with the following addition: each example belongs not only to its class, but also to all ancestors of the class in the hierarchical tree, except the root. We exclude the root of the tree, since all examples belong to the root by default. We call our new measure hP (hierarchical precision) and hR (hierarchical recall). Figure 2 shows how we calculate our new measure.

$$hP_i = \frac{TP_i}{TP_i + FP_i} \qquad hR_i = \frac{TP_i}{TP_i + FN_i}$$

Our measure counts the number of correctly predicted categories along with the number of correctly predicted ancestors of these categories assuming that instances also belong to the ancestors of the correct categories. In the picture on the left, both categories, a parent and a child, were classified correctly; therefore, we get perfect precision $hP = 2/2 = 1$ and perfect recall $hR = 2/2 = 1$. In the picture in the middle, an instance is misclassified into a sibling of the correct category. In other words, the parent category is predicted correctly, but the child category is not. Since one of the two assigned categories is predicted correctly, we get precision $hP = 1/2$. Similarly, since one of the two correct categories is assigned by a classifier, we get recall $hR = 1/2$. In the picture on the right, an instance is misclassified into the parent of the correct category. Since only the true category (the parent of the correct category) is assigned, we get perfect precision $hP = 1/1 = 1$. However, only one of the two correct categories is assigned, so we get recall $hR = 1/2$.

4. ADVANTAGES OF THE HIERARCHICAL APPROACH

The hierarchical approach to the problem of automatic gene annotation has several advantages over the conventional "flat" approach.

4.1 Visualization

Visualizing categories assigned to a group of genes of interest in a hierarchical graph can clearly show the relationships among the genes: do they share the same or similar functions, what functions do they share, do they participate in the same or related biological processes, etc. Even if the genes are not responsible for the same process, the hierarchical graph can show how biologically close their functions are. In the case of "flat" classification, assignment of a wrong category can possibly lead a researcher to the conclusion that two genes perform completely different roles in an organism, while in the case of hierarchical classification the researcher would still be able to reveal the genes' relationships if their functions belong to the same subgraph.

4.2 Scalability issue

Gene Ontology consists of hundreds of categories; a typical microarray study deals with hundreds or thousands of genes; Medline library contains over 12 millions abstracts. Complex state-of-the-art learning algorithms would require a lot of computational resources to analyze such a huge number of documents and produce a classifier for numerous relevant features and a vast number of categories. Besides, most of the algorithms are not suited for problems of such scale and would produce classifiers with very poor performance. At the same time, a hierarchical approach allows us to divide a large problem into several smaller subproblems in a divide-and-conquer manner. Each node in a hierarchical graph can be seen as a separate subproblem: only documents belonging to this node are considered as training documents and only categories corresponding to the children nodes of this node are considered as categories. These subproblems have only a few categories; therefore, we can use more complex algorithms on them and get better classification results.

4.3 Additional source of information

A hierarchy carries potentially valuable information about the relationships between the categories. It is an additional source of information that can be incorporated into the learning process and possibly improve the classification

accuracy. For example, in a local hierarchical approach we can perform feature selection at each node of a hierarchy (a subproblem) independently, choosing a small number of features the most relevant for a particular subproblem. The subproblems can differ substantially and, therefore, should be categorized by different terms. Like in the example from [2], words "farm" and "computer" are good indicators for topics "agriculture" and "computers", but they are unlikely to be helpful to distinguish between "animal husbandry" and "crop farming" because "agriculture" is likely to appear in documents of both kinds and "computers" most probably will not appear in any documents. As a result, a hierarchical classifier can reach a better performance than that of a flat classifier using only a few words.

In addition, the hierarchical structure carries the notion of different misclassification costs. Clearly, misclassification into a neighbouring node of the correct category is much more preferable than misclassification to a distant node. Yet, conventional "flat" techniques and evaluation measures do not take into account these costs. Learning algorithms and evaluation measures specifically designed for hierarchical categorization can address this issue. For example, our new hierarchical measure can differentiate between different kinds of errors and, therefore, select a classifier more suitable for hierarchical tasks.

4.4 Precision/recall trade-off

The combined hierarchical measure of precision and recall give us an option to balance between the classification precision and the percentage of data classified at the deeper levels of the hierarchy: to get good precision we keep examples at higher levels, to get good recall we push examples to lower levels. A desired ratio between precision and recall depends on the task at hand. If a researcher wants to get as much information about genes used in the experiment as possible, high recall is needed. On the other hand, if the information is required to be as accurate as possible regardless of its completeness, high precision is the goal.

5. CONCLUSION AND FUTURE WORK

In this paper, we present work in progress on automatic gene annotation from biomedical literature using hierarchical text categorization. The proposed system can be useful on its own as an assistant to database curators or as a part of the gene validation process in a gene expression analysis system. The hierarchical approach helps visualize the classification results in a way more intuitive and clear for biologists to analyze; overcome the scalability issue by dividing a large initial problem into several smaller subproblems; improve the performance by incorporating additional information on category relations; and trade off between classification precision and recall.

This work will be continued by conducting experiments in real-life settings and extending the approach by providing additional background knowledge, such as gene aliases, MeSH terms, etc.

Another direction for future research is incorporating assigned GO codes into clustering. One of the main challenges in the gene expression analysis is including background knowledge to produce more meaningful clusters of genes not only with similar expression profiles, but also with common functionalities. We propose a simple solution that can be integrated into many clustering techniques used in

the gene expression analysis. The idea is to include the gene function information that we get at the classification step as an additional feature for the clustering process. The feature values would be the category values in the GO hierarchy predicted by the classification algorithm, and their similarity would be determined based on the distance between categories in the hierarchical graph. The goal is to produce clusters that are more meaningful and more useful for biologists and bioinformaticists than the ones produced by conventional clustering techniques.

6. ACKNOWLEDGMENTS

7. REFERENCES

[1] M. Ashburner et al. Gene Ontology: Tool for the Unification of Biology. *Nature Genetics*, 25(1):25–29, 2000.

[2] D. Koller and M. Sahami. Hierarchically Classifying Documents Using Very Few Words. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 170–178, 1997.

[3] D. Masys, J. Welsh, J. Fink, M. Gribskov, I. Klacansky, and J. Corbeil. Use of Keyword Hierarchies to Interpret Gene Expression Patterns. *Bioinformatics*, 17(4):319–326, 2001.

[4] S. Raychaudhuri and R. Altman. A Literature-based Method for Assessing the Functional Coherence of a Gene Group. *Bioinformatics*, 19(3):396–401, 2003.

[5] S. Raychaudhuri, J. Chang, P. Sutphin, and R. Altman. Associating Genes with Gene Ontology Codes Using a Maximum Entropy Analysis of Biomedical Literature. *Genome Research*, 12:203–214, 2002.

[6] F. Sebastiani. Machine Learning in Automated Text Categorization. *ACM Computing Surveys (CSUR)*, 34(1):1–47, Mar 2002.

[7] H. Shatkay, S. Edwards, W. Wilbur, and M. Boguski. Genes, Themes, and Microarrays: Using Information Retrieval for Large-Scale Gene Analysis. In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 2000.

[8] R. Stevens, C. Goble, and S. Bechhofer. Ontology-based Knowledge Representation for Bioinformatics. *Briefings in Bioinformatics*, 1(4):398–416, Nov. 2000.

[9] J. Tamames, C. Ouzounis, G. Casari, C. Sander, and A. Valencia. EUCLID: Automatic Classification of Proteins in Functional Classes by their Database Annotations. *Bioinformatics*, 14(6):542–543, 1998.

[10] K. Wang, S. Zhou, and S. Liew. Building Hierarchical Classifiers Using Class Proximities. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pages 363–374, 1999.